Real-time Road Surface Mapping Using Stereo Matching, V-Disparity and Machine Learning

Vítor B. Azevedo, Alberto F. De Souza, Lucas P. Veronese, Claudine Badue and Mariella Berger

Abstract— **We present and evaluate a computer vision approach for real-time mapping of traversable road surfaces ahead of an autonomous vehicle that relies only on a stereo camera. Our system first determines the camera position with respect to the ground plane using stereo vision algorithms and probabilistic methods, and then reprojects the camera raw image to a bidimensional grid map that represents the ground plane in world coordinates. After that, it generates a road surface grid map from the bidimensional grid map using an online trained pixel classifier based on mixture of Gaussians. Finally, to build a high quality map, each road surface grid map is integrated to a probabilistic bidimensional grid map using a binary Bayes filter for estimating the occupancy probability of each grid cell. We evaluated the performance of our approach for road surface mapping in comparison to manually classified images. Our experimental results show that our approach is able to correctly map regions at 50 m ahead of an autonomous vehicle, with True Positive Rate (TPR) of 90.32% for regions between 20 and 35 m ahead and False Positive Rate (FPR) not superior to 4.23% for any range.**

Keywords: Road surface mapping, stereo matching, v-disparity, machine learning, mixture of Gaussians, binary Bayes filter

I. INTRODUCTION

The problem of autonomous passenger car navigation has gained increased research interest, specially after the challenges organized by the Defense Advanced Research Projects Agency (DARPA) [1], [2], [3]. In those challenges, the main goal of the participating teams was to develop autonomous cars capable of navigating across the courses specified by DARPA as a series of waypoints. But these waypoints did not correspond precisely to the roads the autonomous cars were expected to navigate through and there could also be obstacles between them. Therefore, the autonomous cars had to somehow build a map of the traversable and non-traversable areas in front of them.

Several techniques can be used to build a map of traversable and non-traversable areas around a car, but depending on the sensors employed, the maximum range may vary considerably [4]. In this paper, we present a computer vision approach for real-time mapping of traversable road surfaces ahead of an autonomous vehicle that relies only on a stereo camera. Our system first determines the camera position with respect to the ground plane using stereo vision algorithms and probabilistic methods, and then reprojects the camera raw image to a bidimensional grid map that represents the ground plane in world coordinates. After that, it generates a road surface grid map from the bidimensional grid map using an online trained pixel classifier based on mixture of Gaussians. Finally, to build a high quality map, each road surface grid map is integrated to a probabilistic bidimensional grid map using a binary Bayes filter for estimating the occupancy probability of each grid cell. We evaluated the performance of our approach for road surface mapping in comparison to manually classified images. Our experimental results show that our approach is able to correctly map regions at 50 m ahead of an autonomous vehicle, with True Positive Rate (TPR) of 90.32% for regions between 20 and 35 m ahead and False Positive Rate (FPR) not superior to 4.23% for any range.

This paper is organized as follows. After this introduction, in Section II we present our approach for road surface mapping. In Section III we describe our experimental methodology and, in Section IV, we analyze our experimental results. Our conclusions and future work follow in Section V.

II. ROAD SURFACE MAPPING

We developed a computer vision approach for real-time road surface mapping that employs only a stereo camera. Our system operates in three main steps: (i) camera state estimation; (ii) image reprojection to the world coordinate system (WCS); and (iii) road surface grid map building.

The camera state estimation is achieved using only computer vision and a Kalman Filter [5]. For that, we first compute disparity maps from stereo camera raw images using a stereo matching algorithm [6], and, from the disparity maps, we then compute v-disparity maps [7], [8] (see Fig. 3(a)). After that we detect the road profile line in the v-disparity maps. Finally, we employ a Kalman Filter to compute the camera state using visual odometry [9] on the prediction step and v-disparity derived information on the correction step.

The image reprojection to WCS is achieved by inverse perspective mapping [10] from the WCS to the camera coordinate system. For that, we first assume that all WCS locations lie on the ground plane and, using the camera state information, we map these world points to the correspondent image points. Then, we build a bidimensional grid map where each grid cell corresponds to a world point in the WCS associated with a point in the raw image, and set this grid cell with the raw image pixel value (see Fig. 3(d)). The generated bidimensional grid map represents a bird's eye view [11] of the terrain in front of the car.

The road surface grid map building uses a pixel classifier based on mixture of Gaussians. In the learning step, the classifier is trained online using a "*safe window*" [12]— a small region in front of the car where it is expected

Vítor B. Azevedo, Alberto F. De Souza, Claudine Badue and Mariella Berger are with Departamento de Informática, Universidade Federal do Espírito Santo, Vitória, ES, Brazil. Lucas P. Veronese is with Mogai Tecnologia de Informação Ltda., Vitória, ES, Brazil.

that there is image of traversable road (see Fig. 3(c)). The learning step maintains a mixture of Gaussians set that models the color properties of the road in the safe windows. In the classification step, the image regions are classified as drivable or undrivable based on how well the image pixels match the color properties of the learned Gaussians, generating a road surface grid map. Finally, in order to produce a high quality map, the road surface grid map is integrated to a probabilistic bidimensional grid map, using a binary Bayes filter for estimating the occupancy probability of each grid cell.

In the following, we provide a detailed description of the three main steps of our approach for road surface mapping.

A. Camera State Estimation

In most of driving situations, the car height and the angle between the car and the ground plane (the pitch angle) are not constant. This happens because frequently the ground plane is not really a flat surface and because the number of passengers and their weight may change from trip to trip. The human vision system is robust enough to tolerate these variations and allows people to build great representations of the world, or, in the case of interest, great road surface maps. These maps make driving possible and even an easy task for most of human beings.

To allow an autonomous vehicle to build proper representations of the world, an artificial vision system needs to tolerate the variation of the camera state. Although the camera state has many degrees of freedom, here we define the camera state as the following two variables (see Fig. 1):

- h: the camera height with respect to the ground plane surface;
- θ: the camera pitch angle, i.e., the angle between the optical axis of the camera and the ground plane surface.

The other degrees of freedom are not essential for building a map of the road with the precision we were looking for.

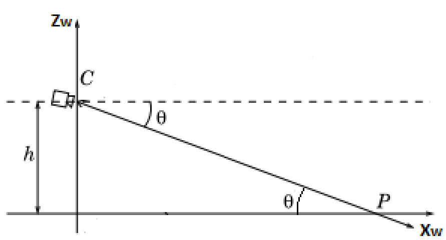


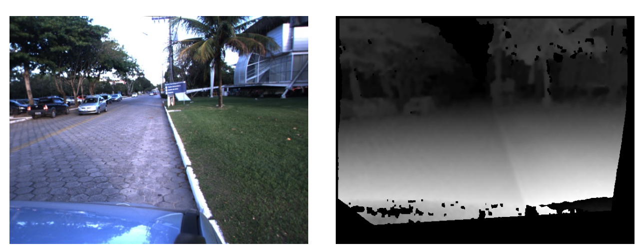
Fig. 1. The camera state and the WCS. The degrees of freedom are the height h and the pitch θ with respect to the ground plane.

Since the camera state is not constant over time when driving, we cannot just measure this state offline and use it to build a map. To solve this problem, we have used the Extended Kalman Filter (EKF [13]), a specific type of Kalman Filter, in order to obtaining a robust approximation of the camera state. A Kalman Filter operates in two steps: prediction and correction. In the prediction step, a mechanism (typically a sensor) is employed to predict the current value of the variables of interest; in the correction step, a different mechanism (typically another sensor) is employed to improve the estimation of the values of the variables computed in the prediction step.

For camera state estimation, we employed visual odometry [9] information on the prediction step and stereo vision derived information on the correction step. The combination of the two is necessary because, on the one hand, the information provided by visual odometry is not precisely correlated to the camera state with respect to the ground plane, but rather with detected features. On the other hand, the information provided by stereo vision is precisely correlated to the camera state, but it is too noisy.

On the prediction step, we calculate variations in the pitch angle and height given by the Library for VISual Odometry 2 (LIBVISO2 [9]). These variations are added to the previous camera state, producing a predicted state.

On the correction step, we use only stereo vision derived information. First, we compute disparity maps from the stereo camera raw images using an OpenMP optimized version of the Library for Efficient LArge-scale Stereo Matching (LIBELAS [6]), that provides good results on outdoor images and also allows real-time processing (see Fig. 2). A stereo disparity map is an image where each pixel value corresponds to the distance in pixels between the image of a point in the world projected in the left and right images of a stereo image pair [11]. Second, from the disparity map, we compute a v-disparity map (see Fig. 3(a)). A v-disparity map is a matrix where each line y corresponds to a line of the disparity map, each column x corresponds to a disparity, and each element (x, y) corresponds to the amount of pixels of line y with disparity x [7], [8]. Third, we detect and extract the parameters of the road profile line from the v-disparity map. The road profile line appears in the v-disparity map because, in typical situations, the road occupies most of the image captured by the stereo camera and, consequently, most of the disparity map (see Fig. 2(b)). Therefore, its profile dominates the v-disparity map—it appears on it as a straight sloped line with an angle greater than $\pi/2$ (see Fig. 3(b)). We detect the road profile line using the OpenCV Computer Vision Library (OpenCV [11]) Hough Transform algorithm, and then extract its slope, c_r, and the image point $(0, v_{0r})$ where the road profile line intersects the vertical axis—from c_r and v_{0r} we can directly estimate θ and h which are used by the EKF correction step.



(a) (b)

Fig. 2. (a) Raw image. (b) The disparity map computed by the LIBELAS stereo matching algorithm[6].

We can directly estimate θ and h from c_r and v_{0r} using Equations 1 and 2 (see [7] for details):

$$\theta = atan(\frac{v_0 - v_{0r}}{\alpha}) \quad (1)$$

$$h = b * \frac{cos\theta}{c_r} \quad (2)$$

where (u_0, v_0) are the image coordinates (only v_0 is used in the equations) of the projection of the optical center of the reference camera (the right camera of the stereo pair) in pixels; α is a derived parameter that is given by $\alpha = \frac{f}{t}$, where f is the focal length of the reference camera in meters and t is the height of a pixel in meters; and b is the distance between the cameras of the stereo pair, also in meters.

B. Image Reprojection to the WCS

Building a bidimensional grid map using range finder sensors such as LIDAR is straightforward, if one knows the sensor position with respect to the WCS [13]. Nevertheless, building a bidimensional grid map using cameras is much more complex, since extracting depth information from the cameras raw image is not easy.

When working with cameras, one starts with a dimensional problem: the WCS is three dimensional, while the cameras' images coordinate system is bidimensional, i.e., the imaging process loses depth information of the real world scene. Stereoscopic vision tackles this problem by adding the disparity dimension to the stereo camera coordinate system. However, stereoscopic vision loses depth resolution as the imaged objects gets further away from the stereo camera [14]. In fact, if the imaged object's distance to the camera is multiplied by $n > 1$, the depth accuracy is decreased by $\frac{1}{n^2}$.

Due to the poor depth accuracy of road regions further than 10 meters of our stereo camera (Point Grey Bumblebee XB3 stereo camera), we have chosen to use the stereo disparity map only to compute the v-disparity map and, from that, to estimate the camera state. Also, we have assumed that all imaged world points are located on the road surface plane. Since most of the world imaged by our car's stereo camera is in fact the road and that we are interested in mapping the road surface, our choices and assumptions makes the most of what our stereo camera can offer us.

To reproject the raw image to the grid map of the road surface plane, we used Equations 3, 4 and 5 for converting from the camera coordinate system, where a bidimensional point is denoted by (x_c, y_c), to the WCS, where a three dimensional point is denoted by (X_w, Y_w, Z_w) (see [15] for details):

$$Z_w = -h \quad (3)$$

$$X_w = \frac{-Z_w * (\alpha * cos\theta + (v_0 - y_c) * sin\theta)}{\alpha * sin\theta + (y_c - v_0) * cos\theta} \quad (4)$$

$$Y_w = \frac{(x_c - u_0) * (X_w * cos\theta - Z_w * sin\theta)}{\alpha} \quad (5)$$

From Equations 3, 4 and 5, one can derive Equations 6 and 7 to convert from the WCS to the camera coordinate system:

$$x_c = u_0 + \frac{\alpha * Y_w}{X_w * cos\theta - Z_w * sin\theta} \quad (6)$$

$$y_c = v_0 + \frac{-\alpha * (X_w * sin\theta + Z_w * cos\theta)}{X_w * cos\theta - Z_w * sin\theta} \quad (7)$$

With Equations 6 and 7, the inverse perspective mapping of the raw image is straightforward and can be described as follows. First, we define a bidimensional grid as the plane at $Z_w = -h$, below the camera optical axis origin in the WCS and in front of the camera. All grid cells of this bidimensional grid have the same size in world coordinates given by r_m x r_m m^2. Then, for each grid cell, we use Equations 6 and 7 to get the raw image pixel that corresponds to the grid cell, and set the grid cell value as the raw image pixel value. Fig. 3(d) shows an example of inverse perspective mapping of the raw image shown in Fig. 3(c) to a bidimensional grid map.

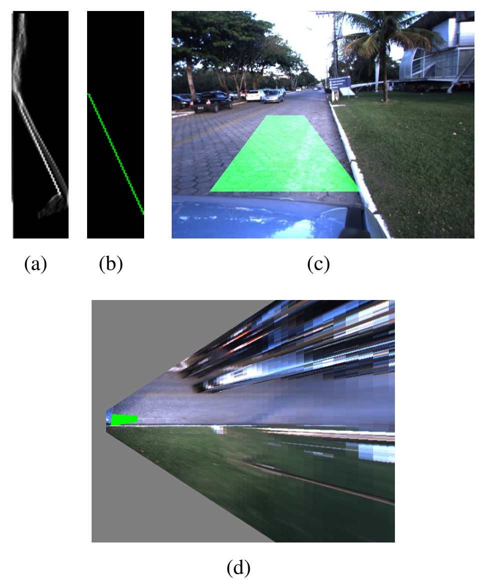


Fig. 3. Inverse Perspective Mapping of the raw image to a bidimensional grid map. (a) V-disparity map computed from the disparity map of Fig. 2(b). (b) Road profile line extracted from the v-disparity map. (c) Raw image with the "*safe window*" highlighted in green. (d) Bird's eye view of the raw image of Fig. 3(c).

C. Road Surface Map Building

Crisman and Thorpe [16] developed a road detection machine learning classifier, named Supervised Classification Applied to Road Following (SCARF), that filters input images and produces output images from which regions can be easily classified as belonging to road surface or not. During the training phase, their algorithm models the road surface color properties with a set of mixture of Gaussians and the non-road surface color properties with another set of mixture of Gaussians. In the classification phase, it computes the probability of each pixel belonging to road surface or not, based on how well the color of the pixel matches the color models.

Thrun et al. [4] used an adapted version of SCARF in their autonomous car Stanley, winner of the 2005 DARPA Grand Challenge [1]. They used a set of Light Detection And Ranging (LIDARs), a Global Positioning System (GPS), a GPS Compass, and an Inertial Measurement Unit (IMU) fixed on top of Stanley to determine the car state, map a world region in front of it, and determine which segments of this region belonged to traversable road surface—a safe window. They also employed a monocular camera for imaging a larger region of the world in front of the car and, employing a 15 degrees of freedom position estimation based on GPS, GPS Compass, IMU, and car odometer, and using straightforward geometric projection, they mapped the safe window into the camera image. In the learning step of Stanley's classifier, for each new camera frame received, the pixels of the safe window are used as new samples to build a mixture of Gaussians classifier. Each multidimensional Gaussian is represented in the RGB color space by a mean color, a covariance matrix and a count of the total amount of image pixels that were used to build the Gaussian. In the classification step, the image is classified using the learned Gaussians in the straightforward way: the image pixels whose value is near one or more of the learned Gaussians are classified as drivable, and all other pixels are classified as nondrivable [4]. But, in contrast to SCARF, Stanley's classifier does not use Gaussians to model non-road surface properties and, at each new camera frame, it may add new Gaussian representations to substitute old Gaussians that are no longer useful or may build new Gaussians and merge them to existing ones to produce more representative Gaussians. With this approach, Stanley was able to expand the LIDARs map from about 22 meters in front of the car to about 70 meters, allowing it to safely drive at higher speeds [4].

We used an approach similar to that of Thrun et al. [4], but with two important adaptations. First, our approach for road surface mapping relies only on a stereo camera that provides RGB frames. In contrast to the Thrun et al. [4] approach that uses several sensors to determine safe windows, we consider a fixed region in front of the car that is more likely to have only road surface pixels (see Fig. 3(c)). But since obstacles could also lie inside the fixed region, we use v-disparity analysis to validate their pixels. With the v-disparity map and the detected road profile line giving information about which pixels belongs to the ground plane (i.e., which pixels are not obstacles), we accept a pixel to compose our safe window if it simultaneously: (i) lies inside the fixed region, and (ii) belongs to the ground plane. Second, our online trained pixel classifier works with the inverse perspective image (instead of the raw one). In contrast to the Thrun et al. [4] approach that determines safe windows and classifies the road surfaces directly on the raw image, we determine safe windows and classify road surfaces on the inverse perspective image (see Fig. 3(d)). This is important because the appearance of the pixels in the raw image varies significantly with their distance from the camera and that does not happen in the inverse perspective image (please, compare Fig. 3(c) and Fig. 3(d) to appreciate that).

Since the classifier is based on color models extracted from safe windows, its learning process can also incorporate information from several kinds of unexpected road with dirty or mud presence. After incorporanting such information from a safe window, the classifier starts classifying dirty and mud as traversable road surfaces. If, after several camera frames, the dirty or mud patterns are not seen again inside a safe window, the learning process will probably replace the old Gaussians that model those patterns with newer Gaussians that model the color properties of recently seen safe windows.

Although our online trained pixel classifier is able to produce a road surface grid map of the region ahead an autonomous vehicle, in order to produce a higher quality map, we employ a binary Bayes filter to integrate each road surface grid map to a probabilistic bidimensional grid map and estimate the occupancy probability of each grid cell. The probabilistic bidimensional grid map is generated by solving the *mapping with known poses* problem using the Occupancy Grid Mapping algorithm [13]. The poses are supplied to the Occupancy Grid Mapping algorithm by visual odometry. Although estimating poses by fusing information from distinct sources (car odometer, GPS, GPS Compass, IMU) is often better, our mapping algorithm works well with the poses provided by visual odometry, which allows our approach to work only with a stereo camera.

Another important aspect of our mapping algorithm is its real-time performance. We developed a Graphics Processing Unit (GPU) implementation able to process images of 640×480 pixels at 20 Hz. This corresponds to a higher frequency than our stereo camera maximum rate of 16 frames per second (FPS).

III. Experimental Methodology

To develop and evaluate our approach for road surface mapping, we have used the open-source Carnegie Mellon Robot Navigation Toolkit - CARMEN [17] (see http://carmen.sourceforge.net). We developed new software modules for some specific sensor controls and stereo vision processing, and also extended the existing software modules for our needs. Using CARMEN modular software architecture and communication patterns, the image processing is distributed across several software modules, that run asynchronously from each other, sharing just the Inter Process Communication - IPC (see http://www.cs.cmu.edu/ ipc/) protocol and some helper libraries. Fig. 4 presents an overview of our road surface mapping system, showing data flow directions and dependencies between the core sensor, drivers and software modules. In Fig. 4, sensors are represented by yellow blocks, sensor drivers by red blocks, stereo vision processing modules by blue blocks and our system core software modules by green blocks.

We decided to build our own dataset using two already available CARMEN facilities: a logger module to record a

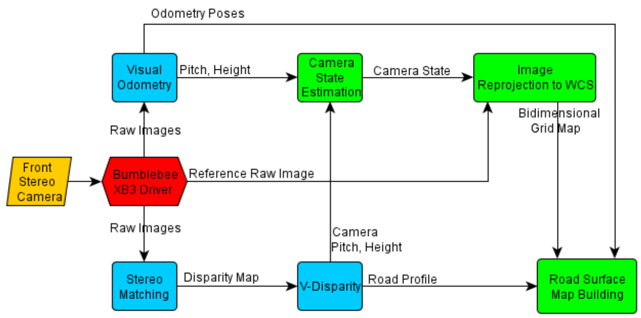


Fig. 4. An overview of our road surface mapping system.

log and a player module to run the generated log offline. The log file is available online at http://www.lcad.inf.ufes.br/log/log_voltadaufes-20120711-1.txt.

In the following, we describe the way our system is assembled, the process to build our logs and the dataset and metris used to evaluate our approach for road surface mapping.

A. System Assembly

To record the logs, we used a common passenger vehicle driven by a human (Fig. 5), with a stereo camera positioned on the top of the vehicle. We have also used a high performance computer with Linux Fedora Core 11 operating system, SSD Hard Disks and RAID 0 technology, to achieve up to 500 MBytes/s of I/O data flow. During the log recording session, the full system (computer and camera) was powered by a single no-break.

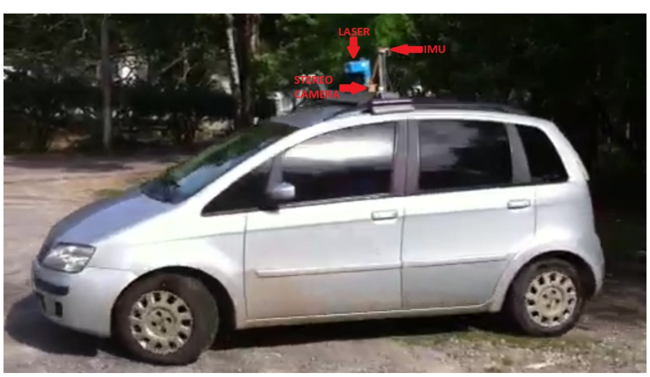


Fig. 5. Lateral view of the experimental vehicle. Assembled on top of the vehicle, a stereo camera, an IMU (not used by our system) and a Laser Range Finder (not used by our system).

The stereo camera used on our experiments was a Point Grey Research Inc. Bumblebee XB3 BBX3-13S2C-38 Stereo Vision Sensor, capturing images of 640×480 pixels at 16 FPS rate, with 66° HFOV. The camera was connected to the computer by a IEEE-1394 firewire interface. The Bumblebee BBX3-13S2C-38 Stereo Vision Sensor datasheet is available online at Point Grey Research Inc. website (see http://www.ptgrey.com/products/stereo.asp).

To test the system, a high performance computer with a GTX-480 CUDA-Enabled NVidia Graphic processing unit with 480 CUDA Cores was used to achieve fast processing rates. Although recommended to real-time execution, there is no need of such a high performance computer to run our system, since we keep both GPU and non-GPU versions of the core software modules, making the system able to run on a computer without CUDA technology.

B. Dataset and Metrics

We selected a set of raw images from our log and the road pixels were then manually classified as drivable. Fig. 6 shows Fig. 2(a) after manual classification. In Fig. 6, the road regions are highlighted by black pixels.

Fig. 6. Fig. 2(a) after manual classification.

We evaluated the performance of our approach for road surface mapping according to two standard metrics for binary classification systems [4]:

- True Positive Rate (TPR), which can be estimated as the ratio between the number of true positives (road pixels that are correctly classified as drivable) and the total number of positives (road pixels).
- False Positive Rate (FPR), which can be estimated as the ratio between the number of false positives (non road pixels that are incorrectly classified as drivable) and the total number of negatives (non road pixels).

While higher TPR is important for not limiting autonomous vehicle planning actions, lower FPR is essential to safely avoid obstacles and non drivable regions.

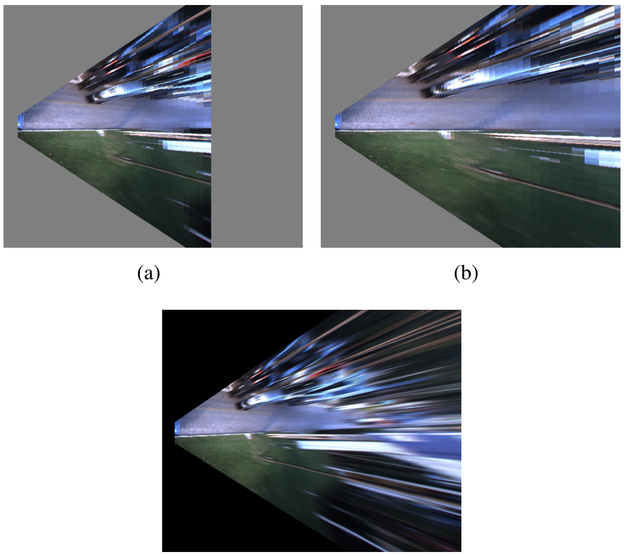
IV. Experimental Results

In this section, we present the experimental results of the performance evaluation of the image reprojection to the WCS and road surface grid map building. A video of the system running is available at http://www.youtube.com/watch?v=-gU3rILw2GQ.

A. Image Reprojection to the WCS

We evaluate image reprojection to the WCS by comparing the results obtained employing EKF (to estimate camera state) with those obtained without EKF. Fig. 7(a), Fig. 7(b) and Fig. 7(c) illustrate our results. Fig. 7(a) shows a poor bidimensional grid map generated without employing EKF, while Fig. 7(b) shows a better bidimensional grid map generated by employing EKF. As Fig. 7(a) and Fig. 7(b) show, a more reliable camera state is able to significantly improve the bidimensional grid map.

We also evaluate image reprojection to the WCS by comparing the results obtained employing EKF with those obtained employing OpenCV Computer Vision Library (OpenCV [11]) bird's eye view method. Fig. 7(c) shows the bidimensional grid map generated by OpenCV. As Fig. 7(b) and Fig. 7(c) show, the bidimensional grid map produced by our approach is more representative of the world ground plane than the one produced by OpenCV.



(a) (b) (c)

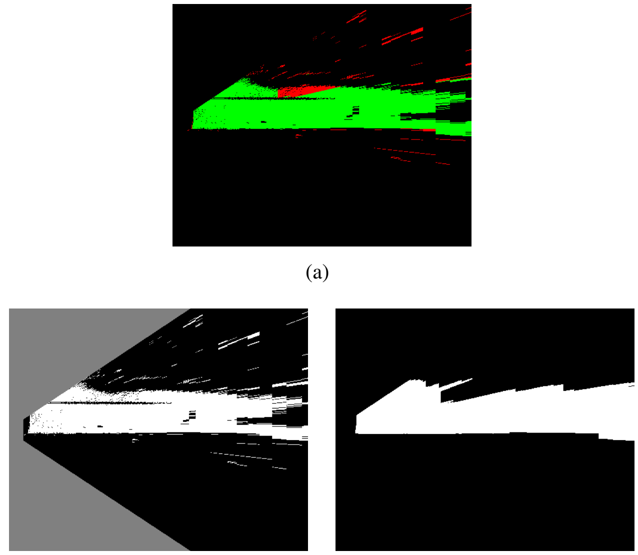
Fig. 7. Bidimensional grid maps (50×40 m). (a) Bidimensional grid map generated without employing EKF. (b) Bidimensional grid map generated by employing EKF. (c) Bidimensional grid map generated by OpenCV.

B. Road Surface Grid Map Building

We evaluate the road surface grid map building by comparing the results obtained by our approach with those obtained by a manually classified road surface map. In order to compare both results at the same basis (i.e., bidimensional grid maps), the set of manually classified raw images from our log were projected to the WCS. In our evaluation, we consider the manual classifier performance results as the ground truth.

Fig. 8(a) shows the performance results, in terms of the true positive and false positive metrics, of our approach in comparison to a manual classifier. In Fig. 8(a), green color pixels highlight true positive regions and red color pixels highlight false positive regions. Fig. 8(b) and Fig. 8(c) show performance results of our approach and those from a manual classifier, respectively. As Fig. 8(a) shows, our approach is able to identify most of the drivable road surfaces, while not incorrectly classifying as drivable most of the obstacles and non drivable regions.

Table I shows the TPR and FPR averaged over the set of raw images from our log at distinct mapping ranges (ranging from near the vehicle to 50 m ahead). Fig. 9 presents the results of Table I in graph form. As Table I and Fig. 9 show, our approach is able to correctly map regions at 50 m ahead



(a)

(b) (c)

Fig. 8. (a) Comparison of the performance results of our approach with those manually classified. (b) Road surface grid map generated by our approach. (c) The bidimensional grid map of the manually classified image shown in Fig. 6.

of an autonomous vehicle, with TPR of 91.86% and FPR of 0.52% for regions up to 10 m ahead; TPR of 81.01% and FPR of 1.66% for regions between 10 and 20 m ahead; TPR of 90.32% and FPR of 1.96% for regions between 20 and 35 m ahead; TPR of 60.89% and FPR of 4.23% for regions between 35 and 50 m ahead. High TPRs over 81.01 for regions until 35 m ahead are important for not limiting autonomous vehicle planning actions, while low FPRs not superior to 4.23% for any region are essential to safely avoid obstacles and non drivable regions.

TABLE I

TPR AND FPR AVERAGED OVER THE SET OF RAW IMAGES FROM OUR LOG AT DISTINCT MAPPING RANGES (RANGING FROM NEAR THE VEHICLE TO 50 M AHEAD).

Range (m)	TPR	FPR
0-10	91.86	0.52
10-20	81.01	1.66
20-35	90.32	1.96
35-50	60.89	4.23

Although not using the same datasets, we can compare the performance results of our approach for road surface mapping with those of previous work by Thrun et al. [4]. Table II shows the performance results, in terms of TPR and FPR, of the approach by Thrun et al. [4]. Our performance results in terms of TPR for regions up to 35m ahead (lines 1-3 column 2 of Table I) are close to that of Thrun et al. in terms of TPR (lines 2-3 columns 2-3 of Table II), and our results in terms of FPR for regions on all ranges up to 50m ahead (lines 1-4 column 3 of Table I) are very close to that of Thrun et al. in terms of FPR (line 6 columns 2-3 of Table II). Fig. 10(a) and Fig. 10(b) show performance results

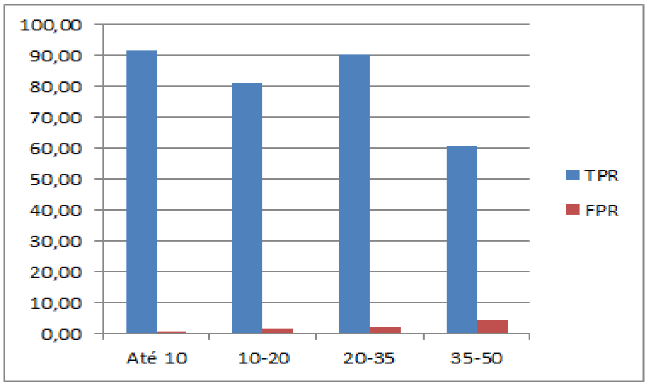


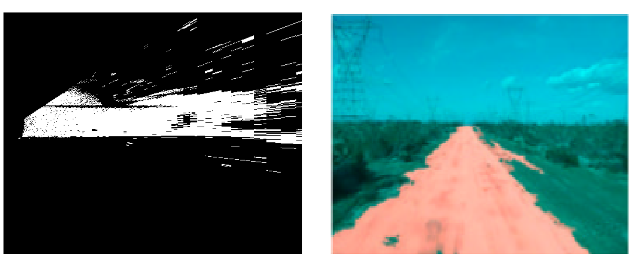
Fig. 9. TPR and FPR averaged over the set of raw images from our log at distinct mapping ranges (ranging from near the vehicle to 50 m ahead).

of our approach and those of previous work by Thrun et al. [4], respectively. In Fig. 10(a), traversable road regions are highlighted in white color and, in Fig. 10(b) non road regions are highlighted in green color. As Fig. 10(a) and Fig. 10(b) show, both approaches are able to correctly classify road surface regions.

TABLE II

TPR AND FPR RESULTS FROM [4] (RANGING FROM NEAR THE VEHICLE TO 50+ M AHEAD).

Range (m)	TPR - Flat desert	TPR - Mountain
0-10	-	-
10-20	90.46	88.32
20-35	91.18	86.65
35-50	87.97	80.11
50+	69.42	54.89
FPR, all ranges	3.70	2.60



(a) (b)

Fig. 10. (a) Performance results of our approach for road surface mapping. (b) Performance results of previous work by Thrun et al. [4].

V. CONCLUSIONS AND FUTURE WORK

We presented and evaluated a computer vision approach for real-time mapping of traversable road surfaces ahead of an autonomous vehicle that relies only on a stereo camera. Using stereo vision (stereo matching and v-disparity analysis), probabilistic methods and machine learning, our approach is able to overcome some difficulties associated with road surface mapping, such as illumination conditions, terrain changes and car oscillation.

We first developed a reliable technique for camera state estimation that employs the Extended Kalman Filter, along with visual odometry information on the prediction step and v-disparity analysis information on the correction step. Our experimental results show that more reliable camera states significantly improve the bidimensional grid maps generated by image reprojection to the WCS. Our results also show that our approach can generate bidimensional grid maps that are more representative of the world ground plane than those produced by OpenCV bird's eye view method.

We also developed a robust way to road surface map building. Our experimental results show that our approach is able to correctly map regions at 50 m ahead of an autonomous vehicle, with TPR of 90.32% for regions between 20 and 35 m ahead and FPR not superior to 4.23% for any range. Although moving obstacles were often presented on the datasets used to evaluate our work, we have not considered them. If we tackle the problem of mapping moving obstacles, our performance results might improve.

A direction for future work is to improve our mapping approach to detect and represent moving obstacles, such as pedestrians and other vehicles, on the environment. We also intend to use the generated road surface grid map to detect road boundaries and estimate lateral offsets of the car with respect to road boundaries. Finally, we plan to use our approach for real-time road surface mapping as part of a solution for autonomous vehicle navigation.

VI. ACKNOWLEDGMENTS

We would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico-CNPq-Brasil (grants 552630/2011-0, 308096/2010-0, and 314485/2009-0) and Fundação de Amparo à Pesquisa do Espírito Santo-FAPES-Brasil (grant 48511579/2009) for their support to this research work. We also would like to thank the research team of Laboratório de Computação de Alto Desempenho (LCAD) from the Universidade Federal do Espírito Santo (UFES) for the technical support and revision of this paper.

REFERENCES

[1] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The Great Robot Race*. Springer, 2005.

[2] M. Buehler, K. Iagnemma, and S. Singh, "Special issue on the 2007 darpa urban challenge, part i," *Journal of Field Robotics, 25(8)*, pp. 423–566, 2008.

[3] M. Buehler, K. Iagnemma, and S. Singh, "Special issue on the 2007 darpa urban challenge, part ii," *Journal of Field Robotics, 25(9)*, pp. 567–724, 2008.

[4] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, and et a, "Stanley: The robot that won the darpa grand challenge." *Journal of Field Robotics 23(9)*, pp. 661–692, 2006.

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering 82*, pp. 35–45, 1960.

[6] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision*, Queenstown, New Zealand, November 2010.

[7] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation," *Intelligent Vehicle Symposium, IEEE*, pp. 646–651, 2002.

[8] F. Lamosa and K. Uchimura, "A complete u-v-disparity study for stereovision based 3d driving environment analysis," *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pp. 204–211, 2005.

[9] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[10] M. Bertozzi, A. Broggi, A. Fascioli, and R. Fascioli, "Stereo inverse perspective mapping: Theory and applications," 1998.

[11] G. Bradski and A. Kaehler, *Learning OpenCV Computer Vision with OpenCV Library*. O'Reilly Media, 2008.

[12] I. Katramados, S. Crumpler, and T. P. Breckon, "Real-time traversable surface detection by colour space fusion and temporal analysis," in *Proceedings of the 7th International Conference on Computer Vision Systems: Computer Vision Systems*, ser. ICVS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 265–274.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[14] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, june 2008, pp. 1 –8.

[15] S. Se and M. Brady, "Ground plane estimation, error analysis and applications," *Robotics and Autonomous Systems*, vol. 39, no. 2, pp. 59 – 71, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889002001756

[16] J. Crisman and C. Thorpe, "Scarf: A color vision system that tracks roads and intersections." *IEEE Transactions on Robotics and Automation, Vol. 9 No. 1*, pp. 49–58, 1993.

[17] M. Montemerlo, N. Roy, and S. Thrun, "Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (carmen) toolkit," in *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003, pp. 2436–2441.